33rd Symposium on Naval Hydrodynamics
Osaka, Japan, October 18-23, 2020

DNS Data Driven Modeling of Turbulent Flows over Rough Walls

M. Aghaei Jouybari, J. Yuan and G.J. Brereton

(Michigan State University, USA)

ABSTRACT

This paper investigates a long-standing question about the effect of surface roughness on turbulent flow: what is the equivalent roughness sand-grain height for a given roughness topography? Deep Neural Network (DNN) and Gaussian Process Regression (GPR) machine learning approaches are used to develop a high-fidelity prediction approach of the Nikuradse equivalent sand-grain height k_s (Nikuradse, 1933) for turbulent flows over a wide variety of different rough walls. To this end, 45 surface geometries are generated and simulated at $Re_\tau = 1000$. The surfaces geometry differ widely in moments of surface height fluctuations, effective slope, average inclination, porosity and degree of randomness. When combined with 15 fully rough experimental data sets (courtesy of Flack et al (Flack et al., 2016; Barros et al., 2018; Flack et al., 2019b)), the DNN and GPR methods predict k_s with an rms error of less than 10% and a maximum error of less than 30%, which appears to be significantly more accurate than existing correlations applied to the present database.

INTRODUCTION

At sufficiently high Reynolds numbers, all surfaces are hydrodynamically rough, as is almost always the case for flows past the surfaces of naval vehicles. Reviews of roughness effects on wall-bounded turbulent flows are provided by Raupach et al. (1991) and Jiménez (2004). The most important effect of surface roughness in engineering applications is an increase in the hydrodynamic drag (Flack, 2018), which is due predominantly to the pressure drag generated by the small-scale recirculation regions associated with individual roughness protuberances.

For the foreseeable future, the most practical approach to making predictive flow calculations at realistic naval operating conditions is to use engineering one-point closure models of turbulence, such as two-equation turbulent eddy viscosity closures for the Reynolds-averaged Navier-Stokes (RANS) equations. Existing rough-wall corrections to this type of closure typically model the increase in hydrodynamic drag on a single length scale—the equivalent sand-grain height (Nikuradse, 1933) k_s—without physically resolving the surface or changing the governing equations. In the fully rough flow regime, where the wall friction depends on the roughness alone and is independent of the Reynolds number, k_s quantifies the increase in hydrodynamic drag through an empirical relation with the roughness function ΔU^+,

$$\Delta U^+ = \frac{1}{\kappa} \ln k_s^+ - 3.5, \tag{1}$$

where $\kappa = 0.41$ is von Kármán's constant, $+$ represents normalization in wall units, and ΔU^+ is defined as the offset of the log-linear velocity profile of a rough-wall flow relative to that of a smooth-wall one, and represents the increase in drag due to surface roughness.

Since k_s is a flow-dependent roughness lengthscale, rather than a physical one, it is not known a priori and does not appear to be equivalent to any single geometrical length scale, such as an average or a root-mean-square (rms) of roughness height (Flack, 2018). It is also well-established that k_s can depend on many geometrical parameters such as the effective slope (Napoli et al., 2008; Yuan and Piomelli, 2014b) and the skewness of the roughness height distribution (Flack and Schultz, 2010). Readers are referred to Flack and Schultz (2010) and Bons (2002) for extensive reviews on this topic. Empirical expressions for k_s based on a small number of geometrical roughness parameters have been proposed in several studies. Examples of existing empirical correlations are

$$k_s = c_1 k_{avg}\big(\alpha_{rms}^2 + c_2 \alpha_{rms}\big), \tag{2}$$

proposed by Bons et al. (2001),

$$k_s = c_1 k_{avg} \Lambda_s^{c_2}, \tag{3}$$

proposed by van Rij et al. (2002) and

$$k_s = c_1 k_{rms} \big(1 + S_k\big)^{c_2}, \tag{4}$$

given by Flack and Schultz (2010). Here, k_{avg} is the averaged height, α is the local streamwise slope angle and $\Lambda_s = \big(S/S_f\big)\big(S_f/S_s\big)^{-1.6}$, where S, S_f, S_s are, respectively, the reference area before adding roughness, the total frontal area of the roughness, and the total windward wetted area. k_{rms} and S_k are rms and skewness of the roughness height fluctuations. c_1, c_2 are constants.

The hydrodynamic lengthscale k_s appears to be correlated with different sets of geometrical parameters for different types of rough surface and no universal correlation currently exists for flow over surfaces of arbitrary roughness. For example, for synthetic roughness comprising closely packed pyramids (Schultz and Flack, 2009) and random sinusoidal waves (Napoli et al., 2008), it has been shown that k_s scales on the effective slope when the surface slope is mild (i.e. within the 'waviness' regime), whereas the skewness and rms height, but *not* slope magnitude, become important when the slope is intense (i.e. within the 'roughness' regime). The boundary between these two regimes has also been shown to be surface dependent (Yuan and Piomelli, 2014b).

In previous studies, the small number of roughness parameters used to devise k_s correlations tended to limit their application to a narrow range of surface roughnesses. Since it appears that many geometrical parameters, such as porosity, moments of roughness height (e.g. rms, skewness and kurtusis), effective slope, and surface inclination angle might affect k_s, it is useful to employ a data science approach suited to modeling large multi-variate/multi-output systems. In particular, we use Machine Learning (ML) to explore k_s-prediction approaches that depend on a large set of surface-topographical parameters, with the expectation that the resulting models may be applied accurately to a wider range of surfaces. The extensive computational and experimental data sets available on k_s, for a wide range of surface roughnesses, are well suited to the requirements of ML.

Since the prediction of k_s from surface topography lengthscales is essentially a 'labeled' regression problem, supervised ML operations were performed, using deep neural networks (DNN), Gaussian process regressions (GPR) and support vector machines (SVM). Readers are referred to the monogram by Géron (2017) and the review provided by LeCun et al. (2015) for in-detail discussions about the methods. All these three ML methods were applied. The major expense in this exercise is the generation and collection of data, rather than evaluating and comparing the performance of various ML procedures per se. The results obtained using the DNN and GPR approaches are reported in this paper; our preliminary effort with SVM did not provide as accurate prediction as the other two approaches. An initial collection of 60 sets of data on k_s as a function of topographical parameters—45 DNS results and 15 experimental results—are considered. All experimental data sets are fully-rough and within DNS data, 30 of them are identified as fully-rough flows and are used for ML training and testing.

The paper is organized as follows. We first describe the governing equations, solution methodologies, simulation parameters and different roughness topographies. Then, the post-processing of DNS results to calculate k_s is presented. Finally, we show the ML model predictions for k_s and discuss the prediction uncertainty.

PROBLEM FORMULATION

Governing equations

The governing equations of incompressible continuity and linear momentum—the Navier-Stokes (NS) equations—for a constant-property Newtonian fluid, were solved using direct numerical simulation (DNS). These equations are written in indicial notation as

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{5a}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + F_i, \tag{5b}$$

Here, $i, j = 1, 2, 3$, x_1, x_2 and x_3 (or x, y, z) are the streamwise, wall-normal and spanwise coordinates, with corresponding velocity components of u_1, u_2 and u_3 (or u, v, w) and P is defined as p/ρ, ρ is the fluid density and ν is the kinematic viscosity. An immersed boundary method (Yuan and Piomelli, 2014a) was used to enforce the fine-grained roughness boundary conditions on a non-conformal Cartesian grid. The corresponding body force F_i is added to the the right hand side of the momentum equations to impose a no-slip boundary condition at the fluid-roughness interface. To solve the equa-

tions, second-order central differencing was used for spatial discretizations and second-order Adams-Bashforth semi-implicit time advancement was employed. The numerical solver was paralleled using the message passing interface (MPI) method (Keating, 2004).

A double-averaging decomposition (Raupach and Shaw, 1982) was used to resolve turbulent and dispersive components of flow variables in the presence of roughness. In this decomposition, any instantaneous flow variable θ may be decomposed into three components, as

$$\theta(\boldsymbol{x},t) = \langle\overline{\theta}\rangle(y) + \theta'(\boldsymbol{x},t) + \widetilde{\theta}(\boldsymbol{x}) \qquad (6)$$

where the time-averaging operator is $\overline{\theta}$ and the intrinsic spatial-averaging operator is $\langle\theta\rangle = \frac{1}{A_f}\int_{x,z}\theta dA$ (and A_f is the area occupied by fluid). The Reynolds and dispersive fluctuating components are then $\theta' = \theta - \overline{\theta}$ and $\widetilde{\theta} = \overline{\theta} - \langle\overline{\theta}\rangle$ respectively.

The calculation of wall shear stress (including both viscous and pressure drag contributions on a rough wall) is by integrating the time-averaged immersed boundary method body force in x, F_1,

$$\tau_w = \frac{\rho}{L_x L_z}\int_{\mathcal{V}}\overline{F_1}(x,y,z)dxdydz, \qquad (7)$$

where $\mathcal{V}$ represents the total simulation domain below the roughness crest and L_{x_i} is the domain size in x_i direction. Readers are referred to Yuan and Piomelli (2014a,c) for details of the implementation and validation of the immersed boundary method and τ_w calculation.

Simulation parameters

Direct numerical simulation was used to calculate the pressure and velocity fields in turbulent open-channel flows over 45 different rough surfaces and one smooth one, at a fixed frictional Reynolds number $\mathrm{Re}_\tau = 1000$. In these simulations, the domain sizes were $(L_x, L_y, L_z) = (3, 1, 1)\delta$, where δ is the channel half-height. The origin of the y axis is set at the elevation of the lowest trough for each rough surface. The number of grid points was $(n_x, n_y, n_z) = (400, 300, 160)$. A uniform mesh was used in the x and z directions, yielding grid sizes of $\Delta x^+ = 7.5$ and $\Delta z^+ = 6.25$, where $+$ denotes normalization in wall units. For all cases, the mesh in y was stretched with a hyperbolic tangent function from the wall onward. The first three y^+ grid points were located below 1. For the rough cases, at the roughness crest, $\Delta y/k_c \leq 0.017$ (with Case C11 giving the highest ratio). Here, k_c is the height of roughness crest measured from the lowest trough of a given surface. The maximum y grid size $\Delta y^+_{\max} = 9.5$ at channel center line (where the Kolmogorov length scale $\eta^+ \approx 3$). Moin and Mahesh (1998) provided arguments that the requirement to obtain reliable first- and second-order flow statistics is that the resolution be fine enough to accurately capture most of the dissipation. Moser and Moin (1987) noted that most of the dissipation in the curved channel occurs at scales greater than 15η (based on average dissipation). Existing DNS studies of channel and boundary layer flows that focus on these flow statistics typically use $\Delta x/\eta$ between 7 and 15 and $\Delta z/\eta$ between 4 and 8 (see, for examples, Kim et al. (1987), Spalart (1988) and Yuan and Piomelli (2014c)). The grid sizes herein satisfied $\Delta x/\eta < 7.5$, $\Delta y/\eta < 4.0$, and $\Delta z/\eta < 6.5$.

The spatial resolution of the roughness Taylor micro-scale, $\lambda_{T,x}$ in x and $\lambda_{T,z}$ in z, are used to evaluate the adequacy of the grid resolution to capture the roughness topographies, according to Yuan and Piomelli (2014a). The Taylor micro-scale is obtained by fitting a parabolic function to the two-point autocorrelation of the surface height fluctuations. It represents the size of an equivalent 'roughness element' in the context of random multiscale roughness. The streamwise and spanwise values of λ_T normalized by δ_ν and the respective grid size are tabulated in Table 1. For all cases, λ^+_{T,x_i} are of orders 10 to 10^2, indicating that average size of the roughness element is large compared to the viscous wall units. The average roughness elements are generally speaking resolved by the grid, with 4 to 12 grid points in each direction per λ_{T,x_i}, for most cases. Yuan and Piomelli (2014b) used a resolution of $\lambda_{T,x}/\Delta x \approx 4$ for the synthesized sand-grain surface (similar to Surface C43 herein) in a large-eddy simulation of channel flow. For some cases here, λ_T is not well resolved in at least one direction (with $\lambda_{T,x}/\Delta x$ or $\lambda_{T,z}/\Delta z$ less than 3). All these cases are among those that are not considered as fully-rough flows (the ones with $\widehat{k}_s < 50$ discussed in the results section) and, consequently, not included in ML training and testing.

In rough wall flows, the pressure drag is caused primarily by the local flow structures and separation in the vicinity of individual roughness protuberances, which are predominately near-wall phenomena. To carry out the 45 separate DNS simulations for determining k_s efficiently, with sufficient near-wall resolution, we employ a small-span channel simulation approach. The concept of minimal-span simulation was introduced by Jimenez and Moin (1991). Chung et al. (2015) and MacDonald et al. (2017) carried out analyses of the perfor-

Table 1: Surface Taylor micro-scale, λ_T, in the streamwise and spanwise directions.

Case name	$\lambda^+_{T,x}$	$\lambda_{T,x}/\Delta x$	$\lambda^+_{T,z}$	$\lambda_{T,z}/\Delta z$
C01,r4,reg,inc1	19.7	2.6	21.1	3.4
C02,r4,reg,inc2	20.4	2.7	33.1	5.3
C03,r4,reg,inc3	19.8	2.6	22.9	3.7
C04,r6,reg,inc1	27.7	3.7	28.4	4.5
C05,r6,reg,inc2	31.6	4.2	39.1	6.2
C06,r6,reg,inc3	29.9	4.0	30.0	4.8
C07,r4,rnd,inc1	33.8	4.5	26.7	4.3
C08,r4,rnd,inc2	26.1	3.5	32.7	5.2
C09,r4,rnd,inc3	35.5	4.7	30.1	4.8
C10,r6,rnd,inc1	38.2	5.1	29.7	4.8
C11,r6,rnd,inc2	38.1	5.1	47.0	7.5
C12,r6,rnd,inc3	47.9	6.4	40.2	6.4
C13,r4,reg,por1	17.8	2.4	32.7	5.2
C14,r4,reg,por2	27.5	3.7	34.2	5.5
C15,r4,reg,por3	31.5	4.2	39.4	6.3
C16,r6,reg,por1	25.6	3.4	46.1	7.4
C17,r6,reg,por2	40.1	5.3	47.8	7.6
C18,r6,reg,por3	44.4	5.9	54.8	8.8
C19,r4,rnd,por1	32.7	4.4	31.1	5.0
C20,r4,rnd,por2	35.6	4.7	31.3	5.0
C21,r4,rnd,por3	37.4	5.0	34.2	5.5
C22,r6,rnd,por1	44.6	5.9	35.3	5.6
C23,r6,rnd,por2	47.1	6.3	39.7	6.4
C24,r6,rnd,por3	47.1	6.3	44.4	7.1
C25,r4,reg,ES1	89.0	11.9	–	–
C26,r4,reg,ES2	66.5	8.9	–	–
C27,r4,reg,ES3	27.1	3.6	–	–
C28,r6,reg,ES1	90.6	12.1	–	–
C29,r6,reg,ES2	66.8	8.9	–	–
C30,r6,reg,ES3	27.2	3.6	–	–
C31,r4,rnd,SGR	27.8	3.7	25.0	4.0
C32,r4,rnd,RND1	131.2	17.5	54.1	8.7
C33,r4,rnd,RND2	96.3	12.8	42.1	6.7
C34,r4,rnd,RND3	56.4	7.5	22.4	3.6
C35,r4,rnd,RND4	39.5	5.3	15.8	2.5
C36,r4,rnd,RND5	25.1	3.3	11.4	1.8
C37,r6,rnd,SGR	36.5	4.9	31.9	5.1
C38,r6,rnd,RND1	88.5	11.8	72.6	11.6
C39,r6,rnd,RND2	93.8	12.5	35.7	5.7
C40,r6,rnd,RND3	57.0	7.6	22.8	3.6
C41,r6,rnd,RND4	40.5	5.4	15.6	2.5
C42,r6,rnd,RND5	24.5	3.3	11.3	1.8
C43,SG	35.2	6.0	33.5	5.7
C44,TB	132.1	10.4	168.5	13.2
C45,CB	25.7	4.5	25.5	4.4

mance of DNS over small spanwise domains for full and open channel flows on rough and smooth walls and showed that minimal-span simulations captured the essential near-wall dynamics and yielded accurate computations of wall friction, and of mean velocities and Reynolds stresses as far from the wall as $y \approx 0.3\delta$, when the following constraints were met:

$$L_x \geq \max\left(1000\delta_\nu, 3L_z, \lambda_{r,x}\right), \tag{8a}$$

$$L_y \geq k_c/0.15, \tag{8b}$$

$$L_z \geq \max\left(100\delta_\nu, k_c/0.4, \lambda_{r,z}\right), \tag{8c}$$

where k_c is the roughness crest height, $\delta_\nu = \nu/u_\tau$, and λ_{r,x_i} is the characteristic roughness wavelength in the x_i direction. The roughness Taylor microscale may be used as the characteristics wavelength; such length is much smaller compared to the domain size for all cases. Conditions (8*a,c*) were satisfied by choosing domain sizes L_x^+ and L_z^+ of 3000 and 1000 respectively. Condition (8*b*) is satisfied for all rough cases except for C11, which falls marginally below one constraint ($L_y = k_c/0.17$).

Periodic boundary condition were imposed in the streamwise and spanwise directions. No-slip and symmetry boundary conditions were imposed at the bottom and top boundaries respectively. When each simulation had reached statistical stationarity, statistics were collected over 5 large-eddy turn-over times (δ/u_τ). The time step $\Delta t^+ \leq 0.04$, orders of magnitude smaller than the Kolmogorov time scale, typically $\tau^+ \approx 2$ in the sublayer of a channel flow according to Choi and Moin (1994).

Surface roughness

In Figure 1, the 45 roughness geometries used for the present DNS are displayed. Each case name in Figure 1 consists of a letter denoting computational ("C") or experimental ("E") data and an index assigned to each surface. For computational cases, the names also have a lengthscale (a percentage of δ) used in generation of that surface, the regular ("reg") or random ("rnd") type of the surface, and the feature that was varied to create that particular surface as one of a series. These features were: the streamwise inclination angle (I_x) in surfaces C01 to C12; the porosity (P_o) in surfaces C13 to C24; and the streamwise effective slope (E_x) in surfaces C25 to C30. For experimental cases two sets of other numbers are assigned to each surface. The first one denotes the year of the publication that the surface was used in; surfaces with number 16 are from Flack et al. (2016), with number 18 are from Barros et al. (2018) and with number 19 are from Flack et al. (2019b).

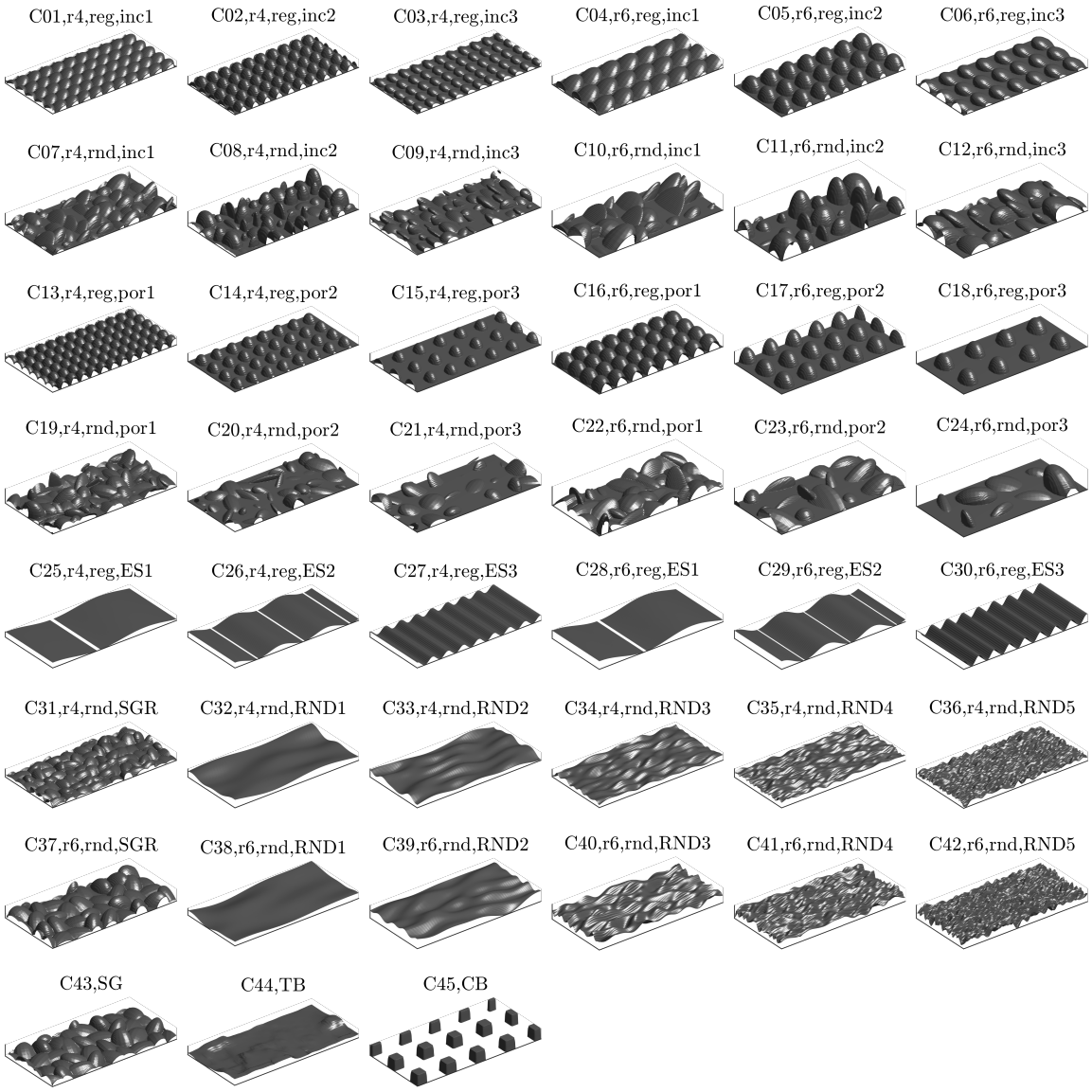

Figure 1: Roughness geometries.

The second number is the index that the surface have been assigned in the original paper.

Surfaces C01 through C24 were created using ellipsoidal elements (Yuan and Piomelli, 2014a) of different size, aspect ratio and inclination. For *regular* roughness, each element has the same orientation and semi-axis lengths, $(\lambda_1, \lambda_2, \lambda_3) = (1.0, 0.7, 0.5)k_c$, following Yuan and Piomelli (2014a). For *random* roughness, the elements have random orientations and semi-axis lengths (with uniform distributions of the random variables). The average orientation and semi-axis lengths are the same as the corresponding *regular* surface. Surfaces C25 through C30 comprise sinusoidal waves in the x direction, of the same magnitude but different wavelengths to generate different values of effective slope E_x. The wavelengths were $\lambda_{r,x} = 3\delta/4,\ 3\delta/8$ and $\delta/6$. Surfaces C31 and C37 comprise the random sand-grain roughness of Yuan and Piomelli (2014a), which were produced by randomly oriented ellipsoidal elements with fixed semi-axes of $(1.0, 0.7, 0.5)k_c$. Surfaces C32 through C36 and C38 through C42 were generated as the low-order (the first 5, 10, 20, 30 and 50) modes of Fourier transforms of white noise in the streamwise and spanwise directions; they therefore describe random surfaces with large to small wavelength roughness. Cases C43, C44 and C45 are DNS (full-span channel) of flow over a random sand-grain roughness, a realistic turbine-blade roughness scan, and arrays of cubes. These surfaces were used in the study of Aghaei Jouybari et al. (2019). Case C46 is the full-span channel simulation of C21, performed to validate the minimal-span channel results. Case C47 is the reference case of a smooth-wall flow.

Other surface parameters calculated for each surface in this work include the mean roughness height (k_{avg}), the first-order moment of height fluctuations (R_a), the root-mean-square of roughness height (k_{rms}), the roughness height skewness (S_k) and its kurtosis (K_u), the effective slope in the x_i direction (E_{x_i}), and the inclination angle (in radians) in the x_i direction (I_{x_i}). The definitions of these geometrical parameters are summarized as the following:

$$k_{avg} = \frac{1}{A_t}\int_{x,z} k dA, \tag{9}$$

$$R_a = \frac{1}{A_t}\int_{x,z} \left(|k - k_{avg}|\right) dA, \tag{10}$$

$$k_{rms} = \sqrt{\frac{1}{A_t}\int_{x,z} (k - k_{avg})^2 dA}, \tag{11}$$

$$S_k = \frac{1}{A_t}\int_{x,z} (k - k_{avg})^3 dA \Big/ k_{rms}^3, \tag{12}$$

$$K_u = \frac{1}{A_t}\int_{x,z} (k - k_{avg})^4 dA \Big/ k_{rms}^4, \tag{13}$$

$$E_x = \frac{1}{A_t}\int_{x,z} \Big|\frac{\partial k}{\partial x}\Big| dA, \tag{14}$$

$$E_z = \frac{1}{A_t}\int_{x,z} \Big|\frac{\partial k}{\partial z}\Big| dA, \tag{15}$$

$$P_o = \frac{1}{A_t k_c}\int_0^{k_c} A_f dy, \tag{16}$$

$$I_x = tan^{-1}\left\{\frac{1}{2}S_k\left(\frac{\partial k}{\partial x}\right)\right\}, \tag{17}$$

$$I_z = tan^{-1}\left\{\frac{1}{2}S_k\left(\frac{\partial k}{\partial z}\right)\right\}, \tag{18}$$

where $k(x, z)$ is the roughness height distribution and $A_f(y)$ and $A_t(y)$ are the fluid area and total area in (x, z) at each y plane. Although the surface geometries were generated using specialized shapes, these descriptive parameters are quite general and applicable to any rough surface. For all surfaces, $R_a/\delta \leq 0.04$, $k_c/\delta \leq 0.17$. The range of values of each parameters can be found in Figure 3.

RESULTS

Post-processed results

In Figure 2, the streamwise double-averaged velocity profiles computed in the present simulations are shown. The profiles in the logarithmic region are described for smooth- and rough-wall cases in the fully rough regime as:

$$\langle\overline{u}\rangle^+ = \frac{1}{\kappa}\ln(y^+) + 5.0, \text{ and} \tag{19a}$$

$$\langle\overline{u}\rangle^+ = \frac{1}{\kappa}\ln\left(\frac{y-d}{k_s}\right) + 8.5 \tag{19b}$$

respectively, where d is the zero-plane displacement, obtained as the location of the centroid of the wall-normal averaged drag-force profile (Jackson, 1981). The shift in the y coordinate by d accounts for the flow blockage by surface roughness elements.

To determine whether a particular flow is within the fully rough regime, Equation (19b) is applied to the computed logarithmic velocity profile to yield a test k_s value, denoted as $\widehat{k}_s$. Its values for all cases are tabulated in Table 2. Cases with $\widehat{k}_s^+$ greater than a threshold value of 50 are deemed to be in the fully rough regime (30 surfaces) and k_s is set to equal

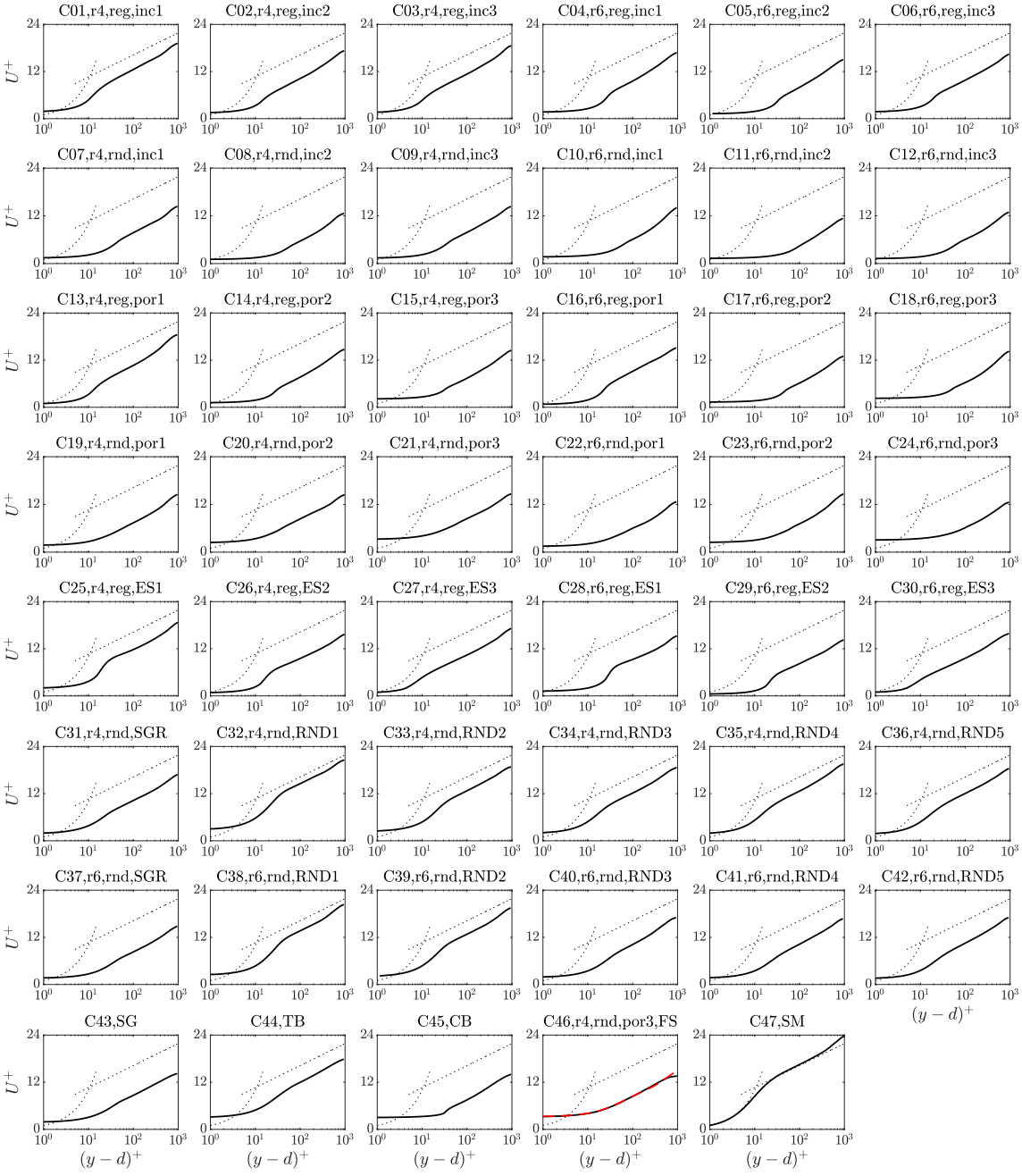

Figure 2: Profiles of streamwise double-averaged velocity plotted against a zero-plane-displacement shifted logarithmic y abscissa. The dashed lines are $u^+ = y^+$ and $2.5 \ln (y-d)^+ + 5.0$. The red dash line in C46 is that of C21.

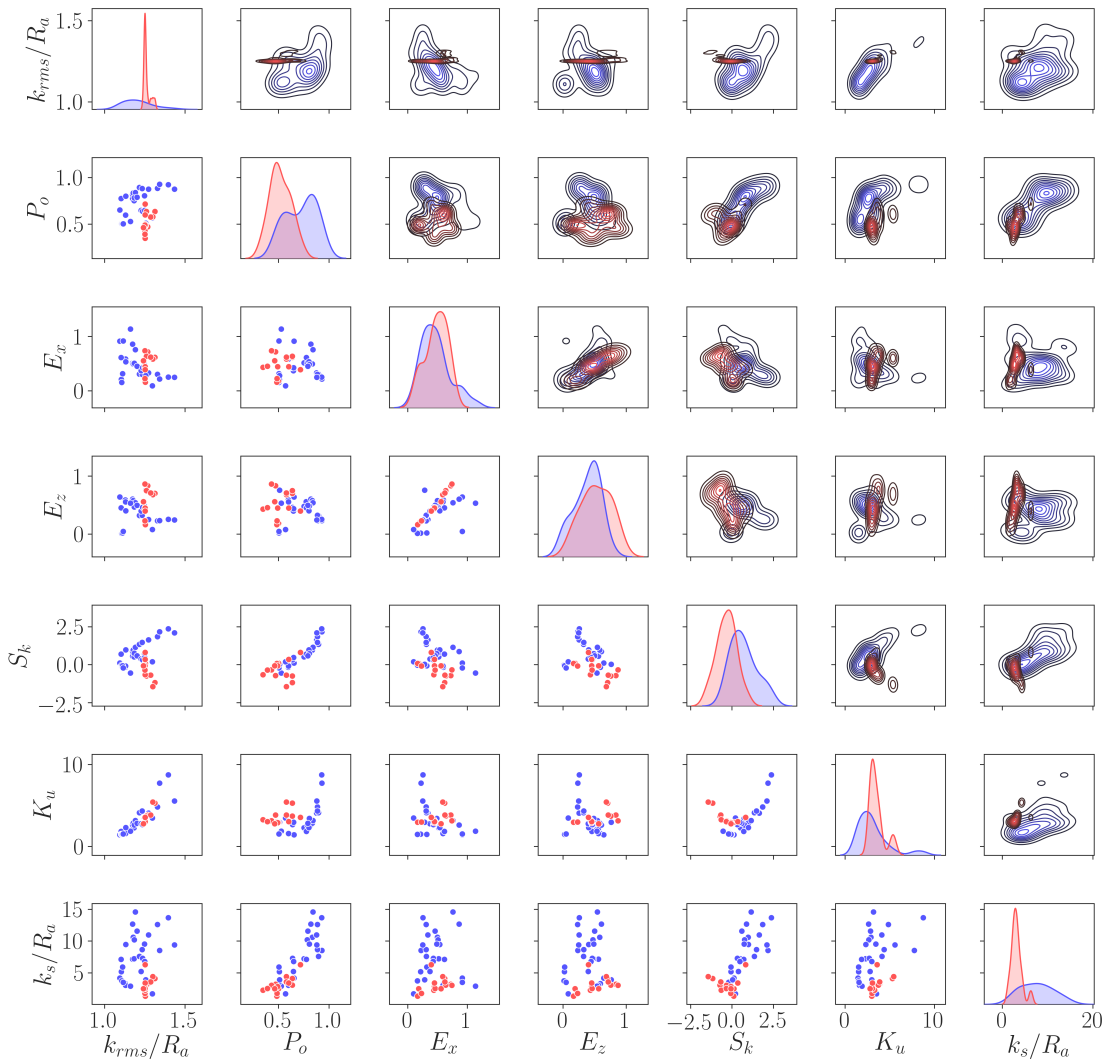

Figure 3: Pair plots of geometrical parameters and k_s, with k_s plots in the bottom row and the first column. DNS data (*blue*), experimental data (*red*).

$\widehat{k}_s$. Those below the threshold are considered transitionally rough (15 surfaces) and are not included in ML predictions in this study. The threshold value of k_s^+—the lower end of the fully rough regime—has been observed to vary significantly for different types of roughness and is typically between 20 and 80. The values of d is also given in Table 2.

The threshold value of k_s^+ which signifies the beginning of the fully rough regime is not determined more precisely because of the cost of carrying out, for each surface, simulations at successively higher values of k_s until the friction coefficient became invariant with Reynolds number. For the GPR method, the uncertainty in k_s which might arise through treating all flows with $k_s^+ > 50$ as fully rough is partially compensated for by incorporating an assumed 10 % noise level of k_s in the learning stage. The values of $k_s^+ = 50$ as the threshold for fully rough flows and the 10% noise level were chosen as a trade-off between the uncertainty in assuming that a given flow is indeed fully rough and the desire to maximize the number of usable data to avoid overfitting. Two other k_s noise levels of 5% and 15% were tested; results suggested insensitivity of the k_s prediction to the noise level within the tested range.

In Figure 3, the pair plots of the different topographic roughness parameters are shown as scatter plots (lower right), joint pdfs (upper left), and distribution pdfs (diagonal). Pair scatter plots for the true (DNS and experimental) value of k_s and other roughness parameters are shown in the bottom row of this figure. It can be seen that, for the roughness cases chosen, there is some correlation between kurtosis and rms roughness (column 1, row 6) and kurtosis and skewness (column 5, row 6) though the relationship between others appears to be more random. From the graphs in the bottom row, it can be seen that k_s/R_a scales on porosity to some power, albeit with some scatter (column 2, row 7). It also appears that k_s/R_a might decrease with skewness for surfaces with $S_k < 0$ and increase with skewness in cases with $S_k > 0$ (column 5, row 7). Surfaces with positive skewness yields higher k_s compared to those with negative skewness, consistent with observation of Flack et al. (2019b). Beyond these observations, there does not appear to be a clear correlation between k_s and any individual roughness parameter, which makes the search for a functional dependence of k_s on these parameters a problem well suited to ML.

Case name	d/δ	$\widehat{k}_s^+$
C01,r4,reg,inc1	0.0316	19.4
C02,r4,reg,inc2	0.0456	49.7
C03,r4,reg,inc3	0.0329	31.0
C04,r6,reg,inc1	0.0379	64.4
C05,r6,reg,inc2	0.0569	124.4
C06,r6,reg,inc3	0.0445	58.9
C07,r4,rnd,inc1	0.0357	136.2
C08,r4,rnd,inc2	0.0522	322.3
C09,r4,rnd,inc3	0.0390	131.1
C10,r6,rnd,inc1	0.0417	268.9
C11,r6,rnd,inc2	0.0702	536.4
C12,r6,rnd,inc3	0.0530	271.7
C13,r4,reg,por1	0.0474	41.4
C14,r4,reg,por2	0.0323	140.6
C15,r4,reg,por3	0.0280	157.1
C16,r6,reg,por1	0.0664	76.7
C17,r6,reg,por2	0.0440	259.8
C18,r6,reg,por3	0.0390	246.5
C19,r4,rnd,por1	0.0417	158.2
C20,r4,rnd,por2	0.0262	105.7
C21,r4,rnd,por3	0.0268	102.7
C22,r6,rnd,por1	0.0528	276.8
C23,r6,rnd,por2	0.0377	175.1
C24,r6,rnd,por3	0.0446	260.3
C25,r4,reg,ES1	0.0235	25.6
C26,r4,reg,ES2	0.0260	65.3
C27,r4,reg,ES3	0.0347	45.5
C28,r6,reg,ES1	0.0334	71.2
C29,r6,reg,ES2	0.0402	112.0
C30,r6,reg,ES3	0.0541	64.0
C31,r4,rnd,SGR	0.0318	48.7
C32,r4,rnd,RND1	0.0410	8.4
C33,r4,rnd,RND2	0.0434	17.6
C34,r4,rnd,RND3	0.0453	22.5
C35,r4,rnd,RND4	0.0463	18.3
C36,r4,rnd,RND5	0.0506	23.4
C37,r6,rnd,SGR	0.0455	108.8
C38,r6,rnd,RND1	0.0602	12.0
C39,r6,rnd,RND2	0.0617	17.1
C40,r6,rnd,RND3	0.0701	50.4
C41,r6,rnd,RND4	0.0729	48.7
C42,r6,rnd,RND5	0.0758	43.8
C43,SG	0.0440	93.0
C44,TB	0.0581	24.1
C45,CB	0.0393	149.9
C46,r4,rnd,por3,FS	0.0270	104.2

Table 2: Flow-related parameters obtained from DNS. The flow is assumed fully rough if $\widehat{k}_s^+ \gtrsim 50$, in which case k_s is equal to $\widehat{k}_s$.

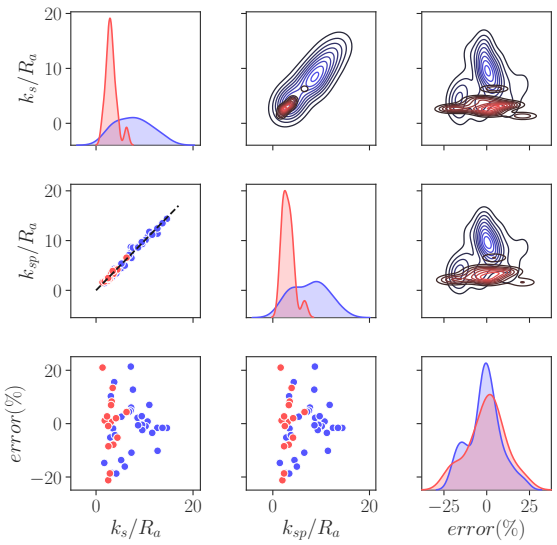

Figure 4: Pair plots of true k_s, predicted k_s (denoted as k_{sp}) and the relative error. DNN method. DNS data (*blue*), experimental data (*red*).

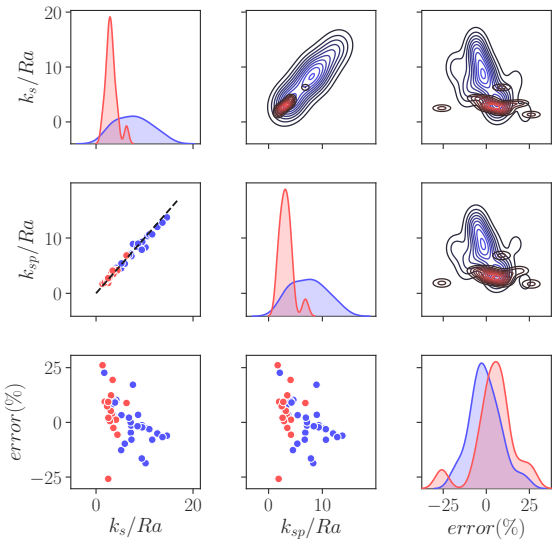

Figure 5: Pair plots of true k_s, predicted k_s (denoted as k_{sp}) and the relative error. GPR method. DNS data (*blue*), experimental data (*red*).

ML predictions of the equivalent sand-grain height

The ML techniques of Deep Neural Networks and Gaussian Process Regression were employed to predict k_s from the data sets described in the previous section. Their main characteristics are described below. The preset parameters to both algorithms were optimized based on available data.

The DNN architecture was a *Multi Layer Perceptron*, with three hidden layers (with 19, 14 and 8 neurons respectively). The activation functions at all nodes were of the *Rectified Linear Unit* kind, and kernel regularization was used to avoid overfitting. The network had 521 trainable weights in total. This architecture was found to provide suitable accuracy in modeling without overfitting, for this particular multivariate labeled regression problem. A *hyperparameter tuning* process were performed. Specifically, 270 configurations were first generated with different lengths (representing the number of layers) and widths (representing the number of neurons). For each configuration, the DNN compiler was performed 1000 times with random selections of training (70% of total) and testing (30% of total) datasets to identify the best performance of the configuration. The configuration that yielded the best results was considered as the optimal one, the results of which are presented here. The cost of data fitting for one iteration (out of 1000) for each DNN configuration was about one second. In total, it took about 75 hours to obtain the optimal DNN network.

The Gaussian Process Regression used *Rational Quadratic* kernels, and incorporated an assumed 10% noise level in the learning stage, for reasons explained in the preceding section. The preset parameters (e.g. kernel type, number of iterations, etc.) were tuned with the available data by running the GPR compiler for about 8000 times. It took about 35 hours to obtain the optimal fit. The GPR method has the capability of incorporating uncertainty or noise in the determination of model parameters in the learning stages. Such noise might arise through: numerical and discretization errors; uncertainty in the form and model coefficients of equation (1); the applicability and fitting range of equation (1) (which was deduced from high Reynolds number experiments) to simulations at much lower Reynolds numbers; and the possibility that some of the training data may have been from simulations in which the flow was not quite fully rough. A noise level of 10% in k_s/R_a values was chosen as an upper estimate of the likely uncertainty from these sources. Noise levels of 5% and 15% were also tested, but lit-

tle sensitivity of the k_s prediction was found to the assumed noise level within the tested range.

The database consisted of DNS of turbulent channel flows over 30 different surfaces at $Re_\tau = 1000$, and the results of 15 experimental data sets. All data were considered to be in the fully rough turbulent flow regime. The data set was split randomly into two groups—one comprising 70% of the data sets and the other 30%—for the respective training and testing processes.

The inputs for both techniques were 17 roughness geometrical parameters, 8 of which were the primary variables k_{rms}/R_a, I_x, $|I_z|$, P_o, E_x, E_z, S_k and K_u (defined in Eqns 9 to 18). The other 9 were products of the primary variables, which were added to improve the efficiency of each learning stage. They were $p_1 = E_x E_z$, $p_2 = E_x S_k$, $p_3 = E_x K_u$, $p_4 = E_z S_k$, $p_5 = E_z K_u$, $p_6 = S_k K_u$, $p_7 = E_x^2$, $p_8 = E_z^2$ and $p_9 = S_k^2$. We chose these products owing to the existing knowledge of the importance of these parameters for certain types of roughness (Yuan and Piomelli, 2014b; Flack et al., 2019b).

The predictions of k_s, henceforth called k_{sp}, are shown as pair plots in Figures 4 and 5, for the DNN and GPR methods respectively. Scatter plots of the equivalent sand-grain height k_{sp} predicted from the surface topography parameters and the true value of k_s (column 1, row 2) reveal a tight clustering of data along the $y = x$ diagonal, with only a few outlying points, in both figures. This very high degree of correlation between k_{sp} and k_s implies that both techniques have been applied with equal success to this prediction problem. The error range (column 1, row 3) is less than $\pm 30\%$ for each technique. It is possible that smaller error ranges might be achieved if important roughness parameters have been omitted from the inputs, if more favorable algebraic combinations of the present choice of input parameters were not explored in the two prediction techniques considered, or if more sophisticated ML techniques had been applied. However, the consistency between both the k_s predictions and error bands for two quite different ML techniques suggests that they are both well-suited to this kind of problem, and possibly close to an optimum for this class of ML approach.

The error values (in percent) for the DNN and GPR methods is given in Table 3, together with the error of the empirical relation

$$k_s = 2.91 k_{rms} (2 + S_k)^{-0.284} \qquad (20)$$

proposed by Flack et al. (2016), which is used as a base empirical relation in our study to compare with. Note that in Equation 20 the original coefficient values from Flack et al. (2016) were used; these coefficients were not fitted using the present datasets. For the DNS data set, only the fully-rough cases are compared. Results show that the errors associated with Equation (20) are small for surfaces with index E-16 as they were used to calibrate this relation. For the other surfaces, the errors reach as large as 120%, owing to the small number of training data and geometrical variables used in the calibration of Equation (20).

In the DNN and GPR predictions of simulation cases, the greatest errors (from 15% to 23%) arise in cases C04, C05, C07, C12, C20, C28, C29, C44, and C45. The surfaces associated with most of these cases may be roughly grouped into two categories: i) those characterized predominantly by close packing of roughness elements (C04, C05, C07 and C12), which contribute to flow sheltering (Jiménez, 2004) that reduces pressure drag, and ii) those characterized by low ES_x or sparse distribution of local peaks (C28, C29 and C44). These observations suggests that it may be more difficult to predict k_s accurately for surfaces with closely packed or sparse roughness with the present DNN and GPR methods that were optimized with the present set of inputs.

Although the goal of this work is to offer a general approach for a wide range of different types of roughness, the ML methods can also be used for a subset of similar roughness types (e.g. 2D or 3D roughness, regular or random, or those with or without spanwise heterogeneity) to provide accurate (though not universal) predictions. This, however, is outside of the scope of this paper due to the limited data, as a larger dataset would be needed for each particular surface subset.

Uncertainty estimation

The GPR method provides confidence margins for its prediction of equivalent sand-grain height, as functions of each input parameter. These margins can be very useful for indicating the kinds of surfaces for which additional training data could improve the confidence in predictions. This feature of the GPR approach makes it very attractive for studies of this kind, since DNS and experimental generation of data is expensive.

The confidence intervals determined by the GPR technique are shown as functions of the normalized surface rms roughness height, effective slope, porosity and skewness in Figure 6. Wider intervals indicate higher estimated values of predictive error, such as at roughness porosity of 0.67, and skew-

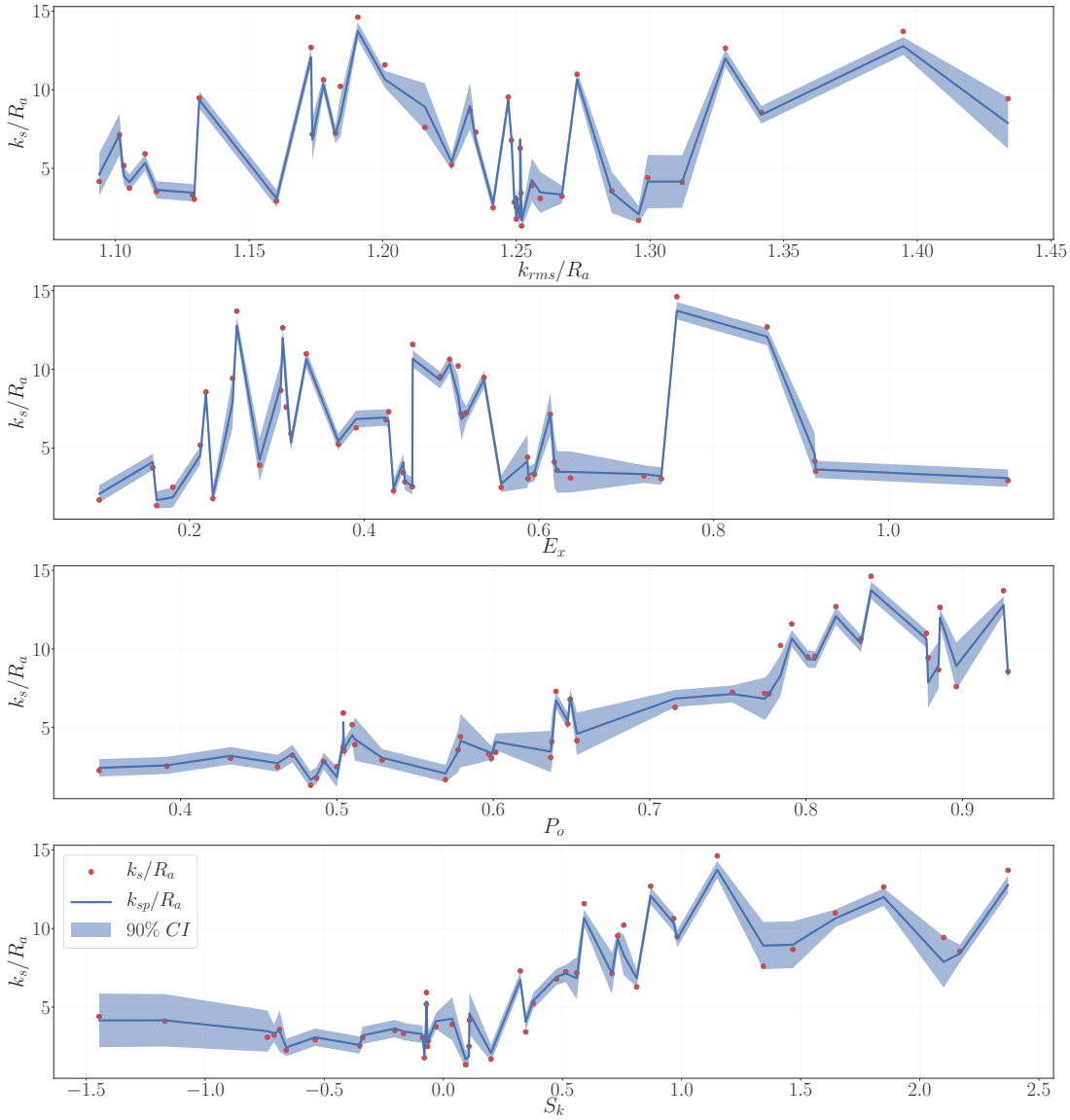

Figure 6: Confidence interval of prediction obtained from GPR method. Predicted k_s/R_a in *blue lines* and true k_s/R_a in *red dots*.

Table 3: Actual errors in k_s prediction for DNN, GPR, compared to err_{base}, errors of the base empirical correlation (Flack et al., 2016) of $k_s = 2.91 k_{rms}(2+S_k)^{-0.284}$. The errors are in percent.

Case name	err_{DNN}	err_{GPR}	err_{base}
C04,r6,reg,inc1	-18.5	4.1	-16.7
C05,r6,reg,inc2	-18.7	10.3	-38.3
C06,r6,reg,inc3	10.3	7.5	-10.4
C07,r4,rnd,inc1	21.4	-4.7	-63.5
C08,r4,rnd,inc2	-10.2	-4.8	-80.1
C09,r4,rnd,inc3	6.0	-1.5	-63.4
C10,r6,rnd,inc1	0.8	-2.7	-76.3
C11,r6,rnd,inc2	-1.6	-6.1	-82.9
C12,r6,rnd,inc3	-1.0	-18.7	-74.7
C14,r4,reg,por2	4.9	0.0	-66.2
C15,r4,reg,por3	7.0	-3.1	-76.7
C16,r6,reg,por1	-5.2	5.4	3.5
C17,r6,reg,por2	-0.4	-1.3	-74.5
C18,r6,reg,por3	-1.6	-5.1	-79.1
C19,r4,rnd,por1	-2.4	-2.3	-71.4
C20,r4,rnd,por2	12.7	17.2	-67.0
C21,r4,rnd,por3	2.2	-1.8	-69.6
C22,r6,rnd,por1	-3.5	-7.9	-77.0
C23,r6,rnd,por2	-0.6	3.4	-70.9
C24,r6,rnd,por3	-1.6	-6.7	-80.5
C26,r4,reg,ES2	-13.7	-12.7	-48.6
C28,r6,reg,ES1	15.6	9.8	-29.2
C29,r6,reg,ES2	-16.1	-9.8	-54.7
C30,r6,reg,ES3	-1.8	3.4	-21.8
C31,r4,rnd,SGR	2.6	3.3	-46.7
C37,r6,rnd,SGR	-10.9	-7.9	-61.3
C40,r6,rnd,RND3	-4.3	9.1	-23.6
C43,SG	4.5	2.1	-58.6
C44,TB	-14.7	22.7	77.6
C45,CB	2.8	-16.5	-70.4
E01,16,2	8.3	3.5	6.2
E02,16,3	0.7	5.2	3.3
E03,16,7	2.0	1.2	-2.2
E04,16,8	-5.3	-5.7	1.3
E05,16,9	7.0	12.4	10.9
E06,16,15	-7.9	-2.5	-3.0
E07,18,1	-21.2	-25.8	17.3
E08,18,2	21.0	26.1	120.7
E09,19,1	1.2	9.4	69.2
E10,19,2	-18.7	0.6	5.8
E11,19,3	2.7	7.4	47.4
E12,19,4	-8.5	2.1	24.1
E13,19,5	13.4	19.4	-16.6
E14,19,6	4.4	8.9	-56.8
E15,19,7	-0.9	9.4	19.8
L_1	7.6	7.8	47.6
L_∞	21.4	26.1	120.7

nesses of -1.2 and 1.5. Surfaces with roughness with these statistical properties are then priorities for further simulations and experiments.

CONCLUDING REMARKS

The equivalent sand-grain height k_s of a rough surface appears to depend on many geometrical parameters. Consequently, methods of prediction based on a subset of only a few parameters have a very limited ability to predict k_s and such predictions can have large deviations from the true value of k_s. While this limitation can be addressed by including more surface-descriptive parameters, it also increases the complexity of the regression procedure needed to determine the predictive model. Machine learning techniques, which can easily handle large multi-parameters problems, can be employed to both predict k_s from multiple inputs and systematically analyze the effect of different geometrical parameters on the prediction. In this paper, we presented a Deep Neural Network and a Gaussian Process Regressor that were shown to be capable of predicting k_s with $err_{rms} < 10\%$ and $err_{max} < 30\%$, which is significantly better than predictions based on conventional analyses. We therefore conclude that these machine learning techniques offer high-fidelity predictions of the equivalent sand-grain roughness height for turbulent flows over a wide range of rough surfaces. These methods are likely to enjoy similar success in related multi-parameter labeled regression problems.

Improved prediction might be achieved by enlarging the database to include rough-wall flows with surface parameters which correspond to low confidence intervals in the GPR method, and by including additional roughness parameters as inputs which might describe sparsity and two-dimensionality, such as frontal solidity, correlation lengthscales and other two-point surface statistics. In addition to k_s prediction described here, the DNS database and the ML techniques in general can also be used to uncover relations between roughness geometry and physics-related quantities, such as the flow pattern around roughness protuberances, flow separation locations, etc. Knowledge of this set of significant roughness parameters may also in turn help build an improved surface database that yields more efficient or more widely applicable predictions of k_s or other quantities.

The machine learning approaches described here are essentially black box techniques, in which the output is a series of weights, or their equivalent, applied at nodes of a network. While such black

box techniques appear to provide efficient and accurate predictions, given a suitable set of training data, they provide little insight into the underlying physical processes, which are usually of great importance for the purposes of design, or extrapolation of predictions beyond the training data. The related problem of deducing a white box model which approximates the predictive capabilities of the black box one, and would reveal information on the features of surface roughness which contribute most to the determination of the hydrodynamic roughness height k_s is a more challenging one, and is a topic for future work.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the Office of Naval Research for the financial support of this research (Award No. N00014-17-1-2102). Computational support was supplied by Michigan State University through computational resources provided by the Institute for Cyber-Enabled Research.

References

M. Aghaei Jouybari, G.J. Brereton, and J. Yuan. Turbulence structures over realistic and synthetic wall roughness in open channel flow at $Re_\tau = 1000$. *Journal of Turbulence*, 20(11-12):723–749, 2019.

J. M. Barros, M. P. Schultz, and K. A. Flack. Measurements of skin-friction of systematically generated surface roughness. *Int. J. Heat Fluid Flow*, 72:1–7, 2018.

J. P. Bons. St and C_f augmentation for real turbine roughness with elevated freestream turbulence. *J. Turbomach.*, 124:632–644, 2002. doi: 10.1115/1.1505851.

J. P. Bons, R. P. Taylor, S. T. McClain, and R. B. Rivir. The many faces of turbine surface roughness. *J. Turbomach.*, 123:739–748, 2001.

H. Choi and P. Moin. Effects of the computational time step on numerical solutions of turbulent flow. *J. Comput. Phys.*, 113:1–4, 1994. doi: 10.1006/jcph.1994.1112.

D. Chung, L. Chan, M. MacDonald, N. Hutchins, and A. Ooi. A fast direct numerical simulation method for characterising hydraulic roughness. *J. Fluid Mech.*, 2015.

K. A. Flack. Moving beyond Moody. *J. Fluid Mech.*, 842:1–4, 2018.

K. A. Flack and M. P. Schultz. Review of hydraulic roughness scales in the fully rough regime. *J. Fluids Eng.*, 132:041203–1—10, 2010.

K. A. Flack, M. P. Schultz, J. M. Barros, and Y. C. Kim. Skin-friction behavior in the transitionally-rough regime. *Int. J. Heat Fluid Flow*, 61:21–30, 2016.

K. A. Flack, M. P. Schultz, and J. M. Barros. Skin friction measurements of systematically-varied roughness: Probing the role of roughness amplitude and skewness. *Flow, Turbul. Combust.*, pages 1–13, 2019b.

Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc., 2017.

P. S. Jackson. On the displacement height in the logarithmic velocity profile. *J. Fluid Mech.*, 111:15–25, 1981.

J. Jiménez. Turbulent flows over rough walls. *Annu. Rev. Fluid Mech.*, 36:173–196, 2004.

J. Jimenez and P. Moin. The minimal flow unit in near-wall turbulence. *J. Fluid Mech.*, 225:213–240, 1991.

A. Keating. *Large-eddy simulation of heat transfer in turbulent channel flow and in the turbulent flow downstream of a backward-facing step*. PhD thesis, University of Queensland, 2004.

J. Kim, P. Moin, and R. D. Moser. Turbulence statistics in fully developed channel flow at low Reynolds number. *J. Fluid Mech.*, 177:133–166, 1987.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

M. MacDonald, D. Chung, N. Hutchins, L. Chan, A. Ooi, and R. García-Mayoral. The minimal-span channel for rough-wall turbulent flows. *J. Fluid Mech.*, 816:5–42, 2017.

P. Moin and K. Mahesh. Direct numerical simulation: a tool in turbulence research. *Annu. Rev. Fluid Mech.*, 30(1):539–578, 1998.

R. D. Moser and P. Moin. The effects of curvature in wall-bounded turbulent flows. *J. Fluid Mech.*, 175:479–510, 1987.

E. Napoli, V. Armenio, and M. De Marchis. The effect of the slope of irregularly distributed roughness elements on turbulent wall-bounded flows. *J. Fluid Mech.*, 613:385–394, 2008.

J. Nikuradse. Laws of flow in rough pipes. *NACA Technical Memorandum 1292*, 1933.

D. Pokrajac, L. J. Campbell, V. Nikora, and I. Manes, C. adn McEwan. Quadrant analysis of persistent spatial velocity perturbations over square-bar roughness. *Exp. Fluids*, 42:413–423, 2007.

M. R. Raupach and R. H. Shaw. Averaging procedures for flow within vegetation canopies. *Bound.-Lay. Meteorol.*, 22:79–90, 1982.

M. R. Raupach, R. A. Antonia, and S. Rajagopalan. Rough-wall boundary layers. *Appl. Mech. Rev.*, 44:1–25, 1991.

M. P. Schultz and K. A. Flack. Turbulent boundary layers on a systematically varied rough wall. *Phys. Fluids*, 21:015104–1—9, 2009.

P. R. Spalart. Direct simulation of a turbulent boundary layer up to $R_\theta = 1410$. *J. Fluid Mech.*, 187: 61–98, 1988.

J. A. van Rij, B. J. Belnap, and P. M. Ligrani. Analysis and experiments on three-dimensional, irregular surface roughness. *J. Fluids Eng.*, 124:671–677, 2002. doi: 10.1115/1.1486222.

J. Yuan and U. Piomelli. Numerical simulations of sink-flow boundary layers over rough surfaces. *Phys. Fluids*, 26:015113–1—015113–28, 2014a.

J. Yuan and U. Piomelli. Estimation and prediction of the roughness function on realistic surfaces. *J. Turbul.*, 15:350–365, October 2014b.

J. Yuan and U. Piomelli. Roughness effects on the Reynolds stress budgets in near-wall turbulence. *J. Fluid Mech.*, 760:R1, 2014c.

Discussion #1

Karen A. Flack, Department of Mechanical Engineering, United States Naval Academy

The paper details the use of a new technique to solve and old problem. Machine learning is utilized to determine the scales of roughness that contribute most to drag. Based on the analysis of 45 fully rough surfaces, the equivalent sandgrain roughness height was predicted and compared to measured or computed values. This is an innovative approach and a welcome contribution to the roughness problem. I look forward to seeing this approach on a wider range of surfaces.

1. Were specific sets of roughness geometries always the outliers (large % difference)? For example, was the technique better for: regular or random roughness, 2D or 3D roughness, roughness without spanwise heterogeneity?

2. What insight would be gained by using the technique for a subset of similar roughness types?

3. What additional surface parameters should be tested? Correlation lengths, solidity, peak-to-trough roughness height?

Authors' reply

Thank you for your comments and questions, each of which is addressed below.

Question 1 Regarding whether there is anything consistent about the surfaces that give high errors, we looked for possible correlations between these errors and roughness parameters or surface categorizations (2D/3D, random/regular) and did not find a clear correlation. However, a subtle trend between relatively high errors and low roughness solidity (or low effective slope, low levels of wake sheltering) was found, with C28 and C44 being two examples. This suggests that the representation of sparse roughness is relatively lacking in the datasets, which should be improved in future works. This point is added to the Results section.

Question 2 Training for specific types of surfaces is likely to improve the k_s prediction for the same amount of samples, since it is equivalent to training the networks for a subset of surfaces that are more similar. The ML methods can be used for this purpose to make non-universal predictions, while the present goal is to predict k_s for a wide range of surface types. It is also out of the scope of this work due to limited data, as a large dataset would be needed for each particular surface subset.

Generally speaking, a ML network trained for one type of surfaces cannot be used for a different type of surfaces. For example, if we divide the dataset into 2D and 3D surfaces, where $E_z = 0$ and $|E_z| > 0$ respectively, we get two different correlations. The one for the 2D case will be independent of E_z, as it is invariant throughout this subset, and the one for the 3D surface would not be accurate for the surfaces with $E_z = 0$, because $E_z = 0$ is beyond the region where the ML is trained on. An extrapolation would not be accurate for ML predictions.

Question 3 Following the discussion on Question 1, additional roughness parameters as inputs which might describe sparsity and two-dimensionality (such as frontal solidity, correlation lengthscales and other two-point surface statistics) need to be added. This point is added in the Conclusions.

Discussion #2

Ugo Piomelli, Department of Mechanical and Materials Engineering, Queen's University

This paper proposes and tests the use of Machine Learning to determine the value of the equivalent sandgrain roughness k_s, for surfaces with arbitrary roughness distributions. The paper is exceptionally good. The study is conceived and carried out extremely well. The application of ML to this type of problem is novel, and provides a very promising tool.

It is very hard to come up with suggestions for improvement (except for the fact that "monogram" should actually be "monograph"). In terms of future discussion, the authors correctly point out that this type of approach does not provide any information on the underlying physics.

1. How do the authors think that its results can be used to improve our understanding of the physics? And, in turn, how would a better understanding of the physics improve the prediction of the model?

Authors' reply

We thank the discusser for the comments and questions, each of which is addressed below.

Question 1 Aside from the k_s prediction, the DNS database and the ML techniques in general can also be used to study relations between roughness geometry and physics-related quantities, such as the flow pattern around roughness protuberances, flow separation locations, characteristics of the shear layers associated with the separation bubbles, the wake sheltering volume, etc. Specifically, an ML network trained to correlate these flow characteristics to the roughness geometry may be an efficient tools to distinguish the sets of roughness geometrical features significant for these characteristics. Knowledge of this set of significant roughness parameters may also in turn help build an improved surface database that yields more efficient or more widely applicable predictions of k_s or other quantities. These are exciting directions of future work based on a successful demonstration of the ML methods for k_s prediction of a wide range of surfaces, which is the goal of the paper. Clarification of the connection between this work and the ML applications in flow physics is now added to the end of the Concluding remarks.

Discussion #3

The 33rd SNH Organizing and Papers Committee (OPC)

This is an interesting article, where the authors try to predict the sand-grain height for fully rough flows using DNN and GPR. They use an extensive DNS database and existing experimental studies to train and develop a framework to predict the sand-grain height.

I would accept that the predictions offered by the two algorithms in general outperforms an existing expression from Flack (2016) so this work clearly highlights the benefits of using ML for closing this problem. However, there are still some things to consider for this article (see my comments throughout the text) and beyond (which the authors mention in the conclusions vis-a-vis the white box model) that would make this a more rigorous study.

1. Comment for line in Abstract "What is the equivalent sand-grain height,...": Does the paper really answer this? From what I can understand, the ML algorithms provide weights to the functions and you don't recover a functional form, similar to what Flack (2016) has. This is clearly a drawback of the ML algorithms chosen. But, can your framework work on a completely different case, not part of either the training or testing data and give a k_s? If so, then you can argue for this statement.
2. Can you expand on what regions these areas refer to?
3. Can you expand on how adding 10% noise compensates for the $k^+ > 50$ assumption? Also, would it then be more rigorous to consider threshold of 80 to only consider fully rough flows? Or does the restriction in the # of usable datasets provide poor prediction of k_s? Have you tested the ML algorithms with different thresholds?
4. Comment for "existing knowledge": Please cite studies that bolster the statement.
5. You might also want to consider modifying the preset parameters to your ML algorithms such as number of hidden layers. If you are providing in the most important input parameters, the algorithm should be able to drive towards a lower error bound in the solution set. Something to investigate before you consider newer ML algorithms.
6. Can you comment a bit more about why for some cases the errors are similar with DNN and GPR but wildly different for others? Consider the second case in Table 3 for the former and eleventh case for the latter.

Authors' reply

We thank the OPC committee for the comments and questions, each of which is addressed below.

Question 1 We agree that the ML algorithms do not recover a functional form for k_s. Clarifications are added in the Conclusions that a white-box approach would reveal information on the surface features which contribute most to the determination of the k_s is a topic for future work. As such, the sentence in the Abstract were rephrased; the word "address" is changed to "investigate".

Question 2 Definitions of these areas are added.

Question 3 The 10% of noise level is incorporated to account for all the uncertainties in data collection. These uncertainties may arise because of numerical error, the application of the equation $\Delta U^+ = 1/\kappa \ln k_s^+ - 3.5$ (deduced from high Reynolds number experiments) to lower-Reynolds-number simulations, the choice of the logarithmic region for the calculation of $\widehat{k}_s^+$, and the possibility that some of the

training data may have been from flows that are not quite fully rough. The manuscript is updated and introduced some of those uncertainties

To evaluate whether the ML prediction errors are primarily due to false-positive detection of fully rough cases with the criterion $k_s^+ \geq 50$, an additional *a posteriori* test was performed for C05, C12, C20 and C28 to evaluate k_s/R_a with a higher Reynolds number ($Re_\tau = 2000$). Large-eddy simulations with the Lagrangian-averaged eddy-viscosity was used, same as Yuan & Piomelli, JoT, 15 (2014). These three roughness geometries were chosen as their predicted k_s gave around 10% - 20% errors for one or both ML methods. Results showed that the $2\times$ higher Re_τ led to 4-5 % difference in k_s/R_a for C05, C12 and C20, indicating that they are fully-rough at $Re_\tau = 1000$. For C28, however, a difference of 25% in k_s/R_a was obtained based on the higher Re_τ. Another two cases, C44 and C45, were previously found to be fully rough at $Re_\tau = 1000$ but gave relatively high ML prediction errors also (23% and 17%). These observations suggest that the inclusion of transitionally rough cases contributed to the k_s prediction error, while it is probably not the main error source as the high-error cases are mostly fully-rough cases.

Based on the above discussions, we consider 10% of noise level as a reasonable trade-off in this work. To evaluate whether the k_s predictions are sensitive to the noise level, we performed the GPR prediction with different noise-levels of 5% and 15%. The k_s prediction error magnitudes and distribution did not differ significantly with that of 10%. This clarification is added to the paper.

We agree that the number of dataset (45) is considered small for a ML approach. As research groups start to share roughness datasets and measurements, we are likely to see growing databases useable for this type of approach in the near future. A limited dataset may lead to overfitting in the present methods. Once more data become available, it will be possible to estimate a minimal test size for good accuracy, using a learning curve method. Constrained by limited data, the following measures were used herein to decrease risk of overfitting: (1) use small number of input features, (2) use as simple neuron networks as possible, and (3) perform kernel regularization for DNN and adding noise level for GPR.

Question 4 Citation added.

Question 5 The preset parameters of both DNN and GPR were optimized based on available data. The following clarifications are added. (1) For DNN, a hyper-parameter tuning process were performed. Specifically, 270 configurations were first generated with different lengths (representing the number of layers) and widths (representing the number of neurons). For each configuration, the DNN compiler was performed 1000 times with random selections of training (70% of total) and testing (30% of total) datasets to identify the best performance of the configuration. The configuration that yielded the best results was considered as the optimal one, the results of which are presented here. The cost of data fitting for one iteration (out of 1000) for each DNN configuration was about one second. In total, it took about 75 hours to obtain the optimal DNN network. (2) For GPR, The preset parameters (e.g. kernel type, number of iterations, etc.) were also tuned with the available data by running the GPR compiler for about 8000 times. It took about 35 hours to obtain the optimal fit.

Question 6 The difference of predictions based on DNN and GPR may be due to the fact that the two methods are based on very different principles. In essence, DNN assigns weights to nodes based on training, while GPR calculates probability distribution based on multivariate Gaussian. In addition, the effectiveness of tuning is likely to vary.